Listing of Claims:

1. (Original) A method for automatically preventing errors in computer software, the method comprising:

storing the computer software in a code repository;

executing a plurality of software verification tools to verify the computer software, wherein each of the plurality of software verification tools has a verification scope and automatically generates one or more test cases;

generating verification results responsive to executing the plurality of software verification tools and the automatically generated test cases;

processing the verification results for generating an objective criterion of quality of the computer software; and

customizing the verification scope of one or more of the plurality of verification tools responsive to the objective criterion of quality of the computer software.

2. (Original) The method of claim 1 further comprising providing a common configuration file for the plurality of verification tools.

3. (Original) The method of claim 2, wherein the step of customizing the verification scope comprises modifying the common configuration file responsive to the objective criterion of quality of the computer software.

4. (Original) The method of claim 2 further comprising modifying a portion of the common configuration file specific to one of the plurality of verification tools responsive to the objective criterion of quality of the computer software.

5. (Original) The method of claim 2 further comprising modifying a portion of the common configuration file specific to one of a plurality of software developers responsive to the objective criterion of quality of the computer software.

6. (Original) The method of claim 1, wherein the step of processing the verification results for generating an objective criterion of quality of the computer software comprises formulating the verification results in a confidence factor represented by the equation:

$$C = p/t \times 100,$$

where p is number of successful test cases and t is total number of test cases.

7. (Original) The method of claim 1, wherein each portion of the computer software being developed by a software developer of a plurality of software developers, and the verification results include a plurality of objective criteria each of the plurality of objective criteria corresponding to quality of a respective portion of the computer software developed by each software developer of the plurality of software developers.

8. (Original) The method of claim 7 further comprising providing a common configuration file for the plurality of verification tools; and modifying the common configuration file responsive to one or more objective criteria corresponding to quality of a respective portion of the computer software developed by each software developer.

9. (Original) The method of claim 7 further comprising verifying a first portion of the computer software developed by a first developer of the plurality of software developers using the plurality of verification tools, before the computer software is stored in the code repository.

10. (Original) The method of claim 9 further comprising allowing storing the first portion of the computer software in the code repository only if result of verification of the first portion meets a set standard.

11. (Original) The method of claim 10 further comprising modifying the set standard responsive to the objective criterion of quality of the computer software.

12. (Original) The method of claim 10, wherein the set standard is common to each of the plurality of software developers.

13. (Original) The method of claim 10, wherein the set standard is unique to at least one of the plurality of software developers.

14. (Original) A system for automatically preventing errors in computer software comprising:

means for storing the computer software in a code repository;

means for executing a plurality of software verification tools to verify the computer software, wherein each of the plurality of software verification tools automatically has a verification scope and generates one or more test cases;

means for generating verification results responsive to executing the plurality of software verification tools and the automatically generated test cases;

means for processing the verification results for generating an objective criterion of quality of the computer software; and

means for customizing the verification scope of one or more of the plurality of verification tools responsive to the objective criterion of quality of the computer software.

15. (Original) The system of claim 14 further comprising means for providing a common configuration file for the plurality of verification tools.

16. (Original) The system of claim 15 further comprising means for modifying the common configuration file responsive to the objective criterion of quality of the computer software.

17. (Original) The system of claim 15 further comprising means for modifying a portion of the common configuration file specific to one of the plurality of verification tools responsive to the objective criterion of quality of the computer software.

18. (Original) The system of claim 15 further comprising means for modifying a portion of the common configuration file specific to one of a plurality of software developers responsive to the objective criterion of quality of the computer software.

19. (Original) The system of claim 14, wherein means for processing the verification results for generating an objective criterion of quality of the computer software comprises means for formulating the verification results in a confidence factor represented by the equation:

$$C = p/t \times 100,$$

where p is number of successful test cases and t is total number of test cases.

20. (Original) The system of claim 14, wherein each portion of the computer software being developed by a software developer of a plurality of software developers, and the verification results include a plurality of objective criteria each of the plurality of objective criteria corresponding to quality of a respective portion of the computer software developed by each software developer of the plurality of software developers.

21. (Original) The system of claim 20 further comprising means for providing a common configuration file for the plurality of verification tools; and means for modifying the common configuration file responsive to one or more objective criteria corresponding to quality of a respective portion of the computer software developed by each software developer.

22. (Original) The system of claim 20 further comprising means for verifying a first portion of the computer software developed by a first developer of the plurality of software developers using the plurality of verification tools, before the computer software is stored in the code repository.

23. (Original) The system of claim 22 further comprising means for allowing storing the first portion of the computer software in the code repository only if result of verification of the first portion meets a set standard.

24. (Original) The system of claim 23 further comprising means for modifying the set standard responsive to the objective criterion of quality of the computer software.

25. (Original) The system of claim 23, wherein the set standard is common to each of the plurality of software developers.

26. (Original) The system of claim 23, wherein the set standard is unique to at least one of the plurality of software developers.

27. (Original) A method for automatically preventing errors in computer software, the method comprising:

executing a plurality of software verification tools capable of automatically generating one or more test cases for verifying the computer software;

processing verification results for producing an objective criterion of quality of the computer software; and

customizing a verification scope of one or more of the plurality of verification tools based on the objective criterion of quality of the computer software.

28. (Original) The method of claim 27, wherein the step of customizing the verification scope comprises modifying a configuration file common to the verification tools based on the objective criterion of quality of the computer software.

29. (Original) The method of claim 28 further comprising modifying a portion of the configuration file specific to one of the plurality of verification tools based on the objective criterion of quality of the computer software.

30. (Original) The method of claim 28 further comprising modifying a portion of the common configuration file specific to one of a plurality of software developers responsive to the objective criterion of quality of the computer software.

31. (Original) The method of claim 27, wherein the step of processing the verification results for generating an objective criterion of quality of the computer software comprises formulating the verification results in a confidence factor represented by the equation:

$$C = p/t \times 100,$$

where p is number of successful test cases and t is total number of test cases.

32. (Original) A method for automatically preventing errors in computer software each portion of the computer software being developed by a software developer of a plurality of software developers, the method comprising:

storing the each portion of the computer software in a code repository;

executing a plurality of software verification tools for verifying the stored computer software, wherein each of the plurality of software verification tools has one or more scope parameters in a configuration file shared by the plurality of software developers, and each of the plurality of software verification tools automatically generates one or more test cases;

generating verification results responsive to executing the plurality of software verification tools and utilizing the automatically generated test cases;

processing the verification results for producing an objective criterion of quality of the computer software; and

changing the one or more scope parameters in the configuration file responsive to the objective criterion of quality of the computer software.

33. (Original) The method of claim 32, wherein the verification results include a plurality of objective criteria each of the plurality of objective criteria corresponding to quality of a respective portion of the computer software developed by a respective software developer of the plurality of software developers.

34. (Original) The method of claim 33 further comprising modifying the configuration file responsive to one or more objective criteria corresponding to quality of a respective portion of the computer software developed by each software developer.

35. (Original) The method of claim 33 further comprising verifying a first portion of the computer software developed by a first developer of the plurality of software developers using the plurality of verification tools, before the computer software is stored in the code repository.

36. (Original) The method of claim 35 further comprising allowing storing the first portion of the computer software in the code repository only if result of verification of the first portion meets a set standard.

37. (Original) The method of claim 36 further comprising modifying the set standard responsive to the objective criterion of quality of the computer software.

38. (Original) The method of claim 36, wherein the set standard is common to each of the plurality of software developers.

39. (Original) The method of claim 36, wherein the set standard is unique to at least one of the plurality of software developers.

40. (Original) The method of claim 34 wherein the step of modifying the configuration file comprises modifying the configuration file by an architect with appropriate access right to edit the configuration file.

41. (Original) A method for automatically preventing errors in computer software written by a plurality of developers, the method comprising:

storing the computer software written by the developers in a code repository;

executing a plurality of software verification tools with respect to the software stored in the code repository using automatically generated one or more test cases, the verification tools having configuration files shared among the developers;

generating verification results responsive to the executed software verification tools and the automatically generated test cases;

processing the verification results to generating an objective criterion of quality of the computer software; and

customizing the scope of one or more of the plurality of verification tools responsive to the objective criterion of quality of the computer software.